A Context Aware Attack Detection System Across Multiple Gateways

Joel Scanlan, Sam Lorimer, Jacky Hartnett, Kevin Manderson

School of Computing
University of Tasmania
Private Bag 100 Hobart TAS 7001

jdscanla@utas.edu.au, sam.lorimer@bigpond.com, J.Hartnett@utas.edu.au, kevin.m@webos.com.au

Abstract

It is well known that intrusion detection systems can make smarter decisions if the context of the traffic being observed is known. This paper examines whether an attack detection system, looking at traffic as it arrives at gateways or firewalls, can make smarter decisions if the context of attack patterns across a class of IP addresses is known. A system that detects and forestalls the continuation of both fast attacks and slow attacks across several IP addresses is described and the development of heuristics both to ban activity from hostile IP addresses and then lift these bans is illustrated. The system not only facilitates detection of methodical multiple gateway attacks, but also acts to defeat the attack before penetration can occur.

Keywords: Intrusion Detection, Firewalls, Prevention, Analysis.

1 Introduction

During the last decade the numbers of networked computers globally has increased astronomically due to the rise of the internet. As a consequence the threat of attack against computer systems is very real, resulting in network security becoming one of the most important priorities not only for system administrators, but for the average network user.

Just as virtually every email user has received spam and virus emails, effectively every computer has had its ports probed, been infected by a virus, or been trivially (or extensively) attacked by another user. While the internet does connect us to the "Information Super Highway" it also allows malicious users to use the same highway t to attack any other user or server connected to it either, directly or through shared network access.

It is at this point that network security infrastructure steps in to provide protection to users from malicious users and their attacks. There are two main types of protective infrastructure which is widely deployed on networks currently: Firewalls and Intrusion Detection Systems. Firewalls act as a means of access control, allowing, and disallowing, access into and out of a given network; Intrusion Detection Systems are designed to detect any malicious behaviour which is occurring on the trusted side of a Firewall. Many networks have grown so large that they often possess several access points to the internet (or to other networks), each protected by an individual firewall or gateway.

This paper will examine a significant shortfall in existing implementations for one of these vital lines of defence. This shortfall is the ability of firewalls to detect or respond to an attack which is being undertaken against multiple gateways on a single network at one time. The reason for this shortfall will be examined, and then methods of responding in an appropriate manner will be explored.

2 Log Analysis

A fundamental component of any system which monitors network activity is the Audit log. The audit log records all of the activities which have happened at a given place within the network; whether it is a gateway or another type of network monitoring sensor. It is this audit log which is examined by Intrusion Detection Systems to identify malicious activity.

Attacks are generally identified by Intrusion Detection Systems examining audit logs by one of the following two methods: Anomaly Detection and Signature Detection.

The Anomaly detection method requires a profile of each user or user group to be made to enable the system to "learn" what comprises normal behaviour (Heberlein et al. 1990; Holden 2003). The behaviour model is then compared to user actions upon the system, searching for behaviour that does not fit the model; this behaviour is then classed as abnormal behaviour and treated as an intrusion.

Anomaly detection is broader than just mapping profiles of human usage. It is also applicable to processes and network access or usage (Hofmeyr, Forrest & Somayaji 1998). Network traffic analysis can yield profiles of normal usage that can be used in monitoring network traffic for anomalies and thus to detect attacks.

The Signature detection method searches audit logs for known attacks, matching malicious behaviour to pre-defined signatures. Signature or misuse detection has a database of attack signatures against which it can compare network event patterns in order to discover an attack. This results in signature detection systems being able to be operational directly after they are installed without the need for any training of the system (Holden 2003).

Signature based intrusion detection is significantly more computationally efficient than anomaly based detection per item of knowledge as it does not need to create matrices for each system activity (Kumar 1995).

However, Brox (2002) comments that signature detection has a flaw in that it requires a signature for a given attack to be able to be detected, and in some instances this is a case of waiting for an attack to occur, to then be able to make a signature to protect against it.

3 Detection Context

The context of network traffic as a topic in network security has grown in importance over the last decade. Initially the focus was only on Firewalls, which have progressed from being simple packet filters to more context aware proxy and dynamic packet filters (Cheswick, Bellovin & Rubin 2003). Likewise, various Intrusion Detection Systems have been designed and modified to also utilize and act upon information based on the wider context of an environment, or users' behaviour (Porras & Neumann 1997; Sommer & Paxson 2003; Vigna, Eckmann & Kemmerer 2000).

There are two primary reasons for the increased usage of contextual information by these two types of network infrastructure: efficiency and accuracy. The initial usage of contextual network information was to decrease the number of packets examined by a firewall; by ignoring packets in an already authorised session. Conversely, Intrusion Detection Systems have tended to use contextual information in conjunction with attack signatures to more accurately detect an attack.

Current Intrusion Detection systems using signature or anomaly detection (or a combination of both) are able to operate effectively on a single gateway or as a network sensor. They analyse audit logs (network traffic) within the context of a single gateway; however if a user attacks multiple gateways upon the same network they are all treated as a single gateway attack at each gateway, meanwhile any breech which does occur is effectively a breech upon them all. An attack which may appear to be trivial at each gateway is in the context of the entire network actually a coordinated attack, and of far greater interest to the network as a whole.

Detection systems attempting to discover attacks which target more than a single gateway on a network need to have a complete knowledge of the given network's context. This entails the system having access to the audit log at each gateway upon the network, and not merely a listing of alerts or threats from detection software which are examining behaviour in a single gateway context. The result is a secondary stage of analysis of network events. This takes place at a central location, examining the logs to look for multiple gateway attacks.

4 Implementation

The work which will be discussed during this paper has been undertaken in two phases, each comprising a year of research as an honour topic. The first phase was aimed at testing the hypothesis that it was possible to detect attacks which occur against multiple gateways upon a single network. The results of this phase will be briefly discussed in this paper, however they have been outlined further in Scanlan et al. (2004). The second phase of the research is currently underway and this paper will discuss a selection of interesting preliminary results from this work.

4.1 Phase 1

The initial work carried out was designed to test the hypothesis that malicious source IP address could be detected attacking multiple gateways upon the same network through centralised analysis. The system which was implemented analysed actual audit data from a gateway range that consisted of multiple remote gateways along with the central server (ns1). Ns1 is bound to the IP addresses of almost a complete C-class running from 0 to 252 in the last octet. The resulting IP range of 'virtual' consecutive gateways, as it appears externally to be 253 separate machines when really each IP address will report to the same machine in one amalgamated log. The audit log still reports which IP address within this range was probed, meaning that it is possible to analyse the data from this single machine as though it were 253 separate gateways. The majority of the log data used in this study is comprised of simple port probes which have been targeted against IP's within the range.

As the goal of the system is to know what is happening across each gateway on the entire network, the retention of context is vitally important. The context of the amalgamated log is preserved through the way in which the log is parsed by the system. The system contains 2 modules: the Analysis module and the Tracking module.

The Analysis module stores a simple profile of each IP which probes the network in a database. This profile consists of such information as IP, target gateway, target port, date and time, ID number, two Boolean values and a probe count. The two Boolean values are used to store whether or not the given user has probed more than one gateway or more than one port. This Audit table effectively compresses the amalgamated log down to being a single entry per malicious IP, indicating how often they have probed the system and whether or not their probes have been against multiple ports or gateways.

The Tracking module examines each IP address at a much greater resolution, in terms of information being recorded, than the Analysis module. The tracking module is not used for every IP, but only those which appear to be of interest to the Analysis module through probing multiple gateways or ports. For each probe which is sent by an IP a new entry is added to the tracking table, recording the IP, target gateway (for each gateway and not merely the first gateway as in the Analysis module), target port, date and time. This allows for the activities of an IP to be closely examined in terms of attempting to define a pattern of attack and this could well be useful for linking different attacking IP's with similar patterns to being the same user who is changing IP addresses.

The second goal of the system was to examine if a threshold level could be established and at which point this should be set. Then if the number of probes from a single user exceeded the threshold they could be classed as a threat to the network as a whole with an acceptable level of accuracy.

	Single Gateways	Multiple Gateways
Phase 1 log		
Source IP Addresses	5990	776
% of Total	88.53%	11.47%
Phase 2 log		
Source IP Addresses	33715	9151
% of Total	78.65%	21.35%

Table 1 Individual Source IP Address

4.1.1 Phase 1 Results

The two main results from the first phase of the research, applied to our project goals: proving the hypothesis that attacks across multiple gateways could be detected, and deciding whether or not an effective threshold level could be established.

As described in section 4, the Analysis module records a series of details about each IP address that probes the network. This includes a count of their actual probes, and whether or not a source IP has probed more than a single gateway. Table 1 displays several statics which were gathered by the Analysis module. The first section of these was gathered on the Phase 1 log file. This log file was about 10MB in size and covered the 10 day study period of the 1st of September 2003 till the 10th of September. During the Study period 6766 individual source IP addresses probed the gateways, of which 776 (11.5%) probed multiple IP addresses. This represents a sizeable risk to networks which had previously gone unnoticed; however noticing that an attack has indeed occurred is only the first step; being able to detect it efficiently is the second half, and the more valued challenge of phase 1.

The analysis module records a count of the number of probes which have been sent by each source IP address against any of the gateways on the network. The second goal of the system was to see if this count could be effectively used to gauge whether or not a user is likely to probe multiple gateways. To investigate the usage of this simple heuristic several test levels were examined. When examining the count value it was found that 83% of source IP addresses sent 3 or less probes against the network. For this reason 3 was the first value we examined as a possible threshold value, followed by 6 and 9 as these were also values where a substantial drop off in probe counts were seen. This heuristic was then used with the Boolean signifying a multiple gateway attack in order to try and classify source IP addresses.

The results showed that at a threshold level of 3 only 3.2% of IP's were classified as potentially performing scans on multiple gateways. With the optimum of 11.5% to get all potential malicious probes it is a relatively poor result. By comparison, when the threshold level was increased to the levels of 6 and 9, the results returned were 8.3% and 10% respectively. These results were much more acceptable, however not quite at the levels desired. Figure 1 illustrates the result from further testing of other threshold levels. The optimum level efficiency was found to be at an 11 probe threshold.

The threshold level signifies the point at which, if a user exceeds the level at a single gateway, they are highly unlikely to probe multiple gateways. The result is that it is possible to track and detect over 90% of users who attack multiple gateways upon a single network.

4.2 Phase 2

Phase 2, which is described in the remainder of this paper, builds directly on top of the system implemented to complete the work in Phase 1.

The first goal in Phase 2 is to verify the work which was done in Phase 1 by using a larger, and longer (in terms of time) log file. Phase 1's log file was not a very large one and this validation is required to certify that the results can be duplicated over a longer time period.

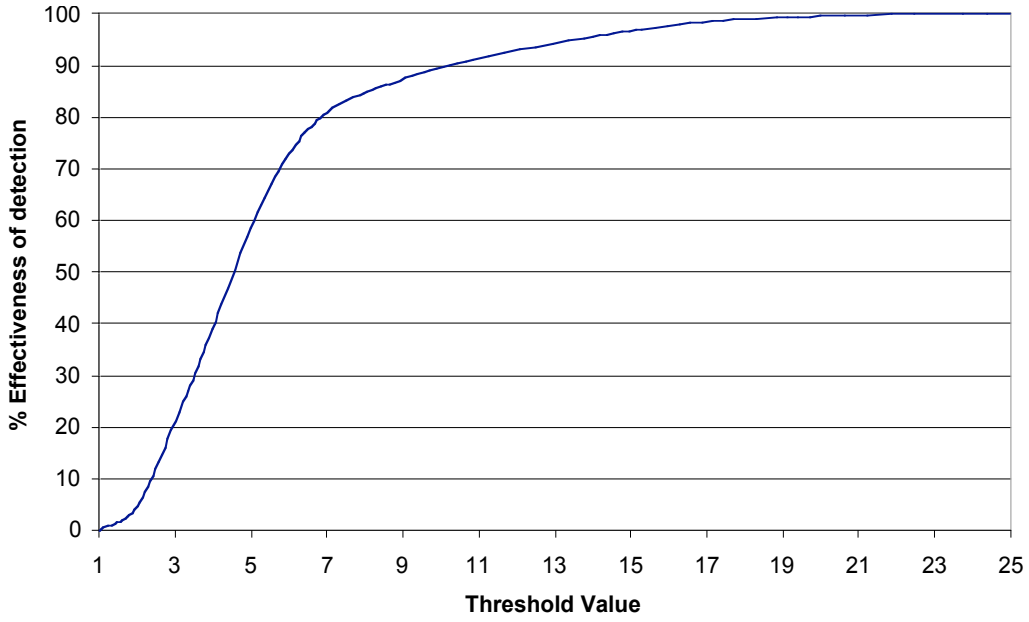

Figure 2. Effectiveness of Detecting IP address probing Multiple Gateways

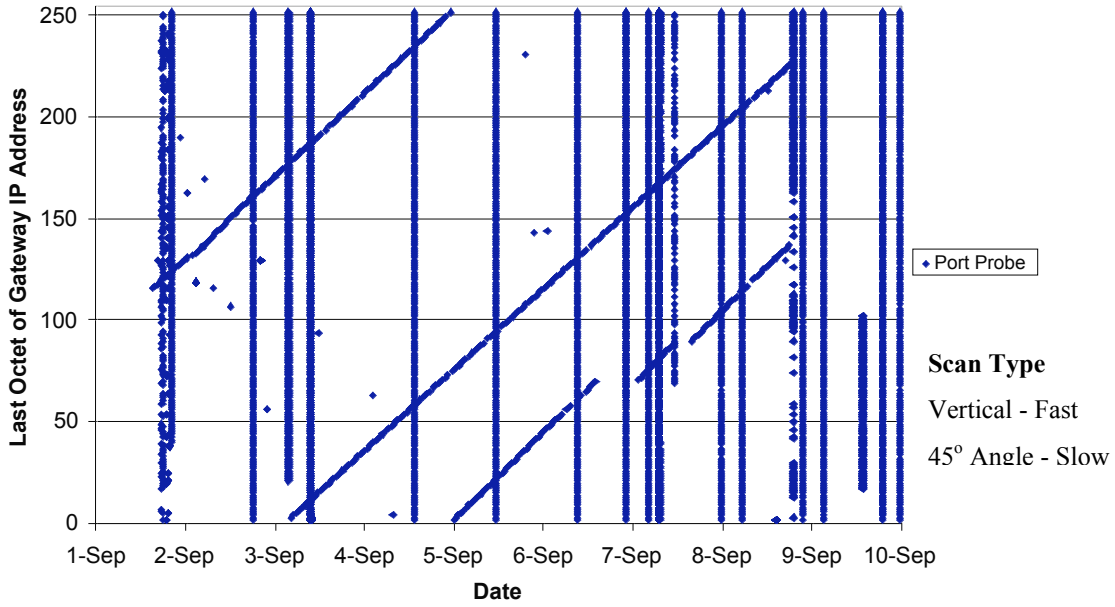

Figure 3 Ten days of gateway activity recorded by the Tracker Module

However, once this is accomplished there are several other more practical goals to examine. One of these is the creation of an action module to respond to the users who attack multiple gateways upon the network. This module needs to protect the gateways on the network which have not been attacked by the malicious user ahead of their future attempted attacks.

The concept of adding additional rules to firewalls ahead of an attack upon the given gateways when a multiple gateway attack may not result in attacks on each gateway is bound to have some administrators sceptical of the system. Concerns about the performance cost of the number of rules which are on firewalls are of great concern to system administrators, and have resulted in rule efficiency applications being produced (Hoffman, Prabhakar & Strooper 2003). Thus the idea of adding extra rules to a series of gateways needs to consider the possible cost in performance on those gateways, as it is obviously of greater concern than when a single rule is added to a single gateway. Phase 2 will aim to be able add and remove the rules it creates in real-time, with the length of the time which a rule is on a firewall to be optimised to be as short as possible, while still providing adequate protection.

For the removal of rules to occur in a timely fashion while still providing protection to the gateways of the network it was necessary to examine the results from the Phase 1 Tracking module to see in what way malicious users were attacking the network. Figure 3 illustrates the two main attacks which were occurring during the Phase 1 log: fast scan and slow scan. The fast scans generally lasted just a few seconds, through to being a few minutes in total length. Figure 4 shows a classic fast example of a scan which lasts 1 min and scans all 250 IP's within the class C address. The slow scans last a far longer, often scanning at similar time intervals between probes lasting.

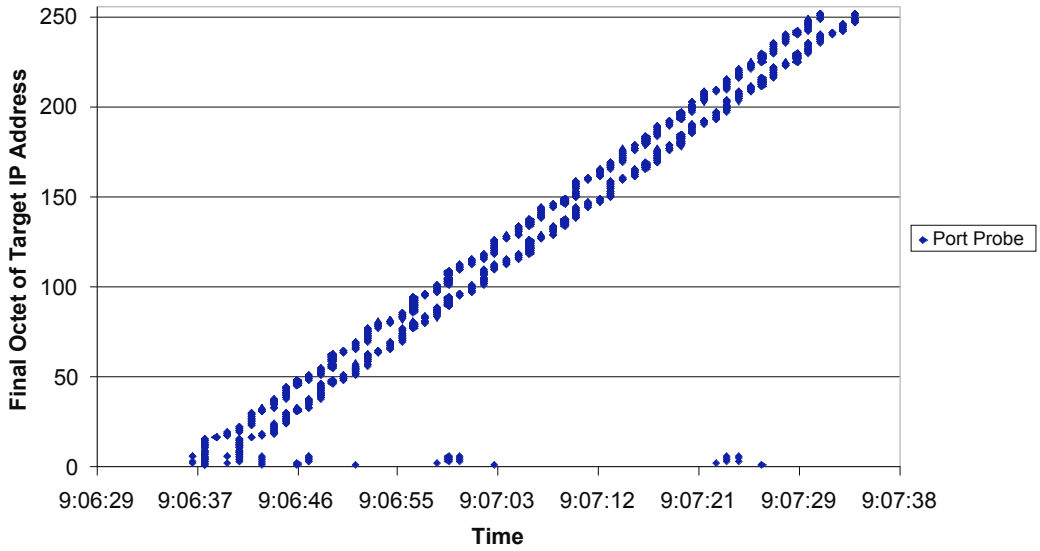

Figure. 4 Scan across multiple gateways from a lone source IP address.

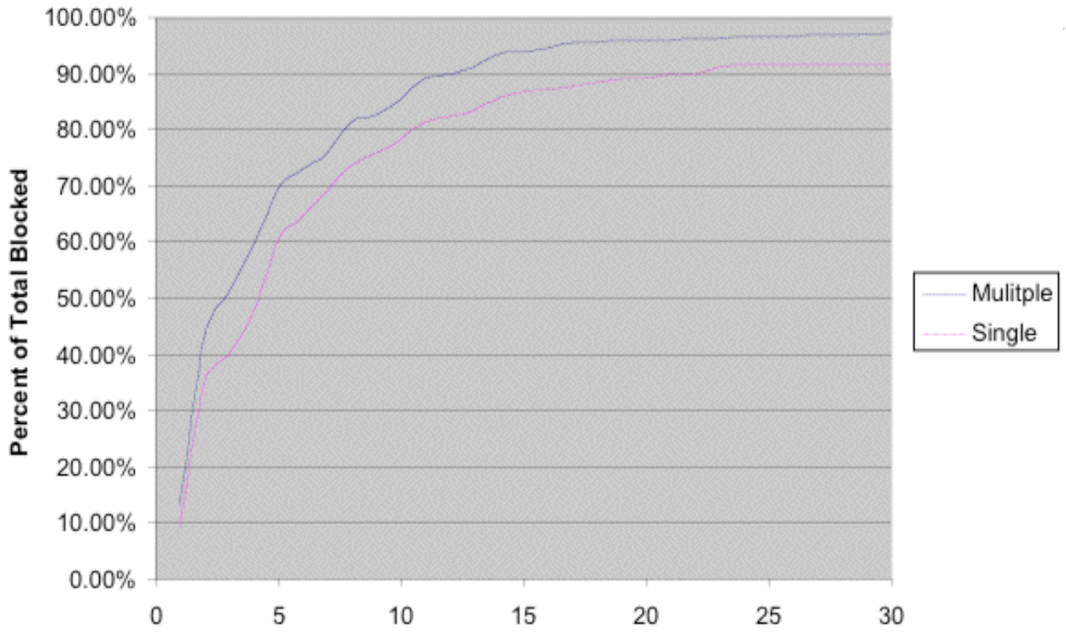

Figure 5 Reaction Time for blocked users across Multiple and Single gateways.

over several days. A third type of scan is a hybrid of the first two types, with a user doing a short fast scan, waiting 24 hours or longer, and then doing a second short fast scan. As a result of the differing scan types, an optimum ban time cannot be a static value and still be effective, it needs to be a dynamic value based on the activities of the given source IP. A further factor affecting the length of the optimum ban time is the apparent difference between malicious users who attack multiple gateways and those who attack single gateways. Figure 5 illustrates the speed at which users stop their activity once they have been blocked upon a gateway. The line which indicates the multiple gateway attackers (the blue, or higher values on the graph) follow is the same general pattern of stopping the attack once blocked, but is substantially faster than that of single gateway attackers. It takes 11 probes for over 90% of the multiple gateway attackers to cease their activities, while it takes twice that at 22 probes for single gateway attackers.

To be able to optimise the ban length without putting the system at risk from a malicious user the Action module also maintains a history of malicious IP's for a short time after their ban being lifted. This will result in the ability to reapply a ban to a repeat offender who returns after their ban has been lifted without them needing to go through the analysis process again, and risk breaking through the firewall and causing any damage to the internal resources of the network.

4.2.1 Phase 2 Results

Phase 2's results can be split into two main groups: Phase 1 validation and automated attack response.

The validation of the results which were produced by Phase 1 was the first priority before adding much further to the system. A few optimisation tests were completed on the system, the results which are displayed in Table 1 (along side those of Phase 1 for comparison). The Phase 2 log is substantially larger than the Phase 1 log, being 270 MB in size, and covering the time period of July 1st through July 21st of 2004. The results showed that not only are multiple gateway attacks also detectable within a larger more comprehensive log; but were actually more prevalent than previously discovered in Phase 1. The Phase 2 log shows that over 20% of probes detected upon the network were from source IP's interested in multiple gateways on the network, a 100% increase on the Phase 1 log. While there is a 10 month difference in time between the logs, it would seem unlikely that they have grown more frequent so rapidly. Further examinations of older log files would be needed to confirm this.

To fully validate the Phase 1 results, the optimum threshold level also needs to be calculated on the Phase 2 log. The Phase 2 log, similarly to the Phase 1 log, also had a large peak at 3 as a possible threshold level. A threshold level of 3 would result in a detection efficiency of above 90%; however that would result in ignoring quite a sizeable number of multiple gateway probing source IP addresses. Figure 6 (over page) clearly illustrates the increasing efficiency levels for different threshold values. The optimum, according to the Phase 2 log results, would indicate that a threshold around 10 would be the most efficient; this validates the threshold of 11 found by the Phase 1 log analysis. Figure 6 also plots the Phase 1 efficiency line for comparison; it clearly demonstrates the differences between the log files, but also their similarity in relation to threshold efficiency.

The remainder of Phase 2 focuses on the way in which the system responds to a source IP which has been found to be probing multiple gateways. The initial task in this was to create an Action module for the system to use to deal with the discoveries from the analysis module. The Action module adds and removes the source IP addresses from the Linux iptable based firewalls running on each of the network gateways. The module also maintains a

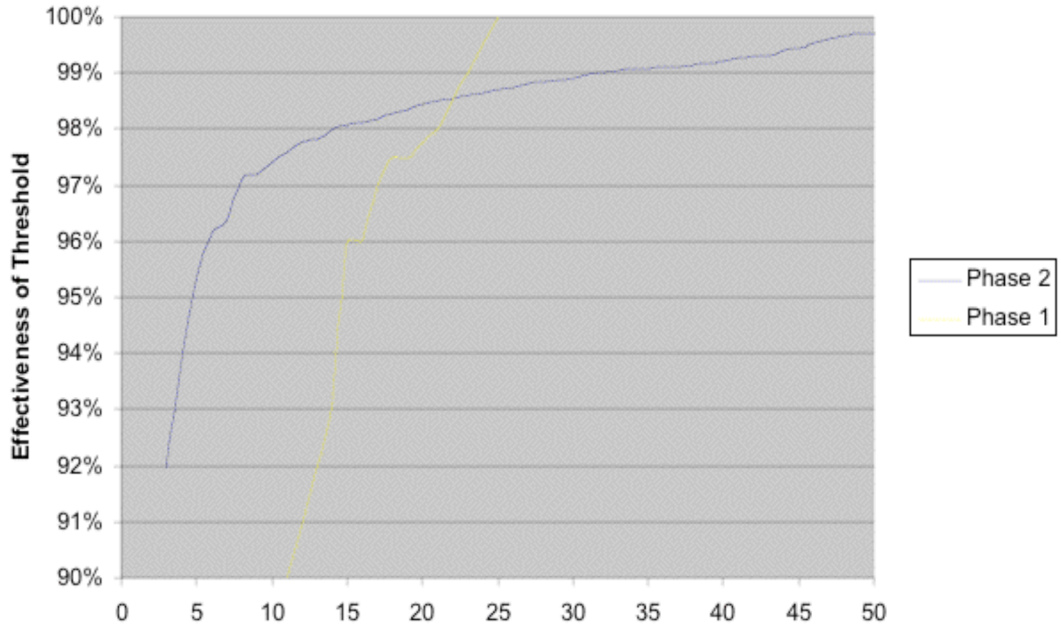

Figure 6 Comparison between Phase 2 and Phase 1 for optimum threshold level.

network state within the database which allows for bans to be lifted upon their expiry.

The concerns mentioned in section 4.2 mean that the length of time which a rule remains in place needs to be as efficient as possible in order to keep the total number of rules to a minimum and preserve network performance. As a result the optimum ban time calculation will need to be both scalable and dynamic. The calculation will need to be scalable to enable it to work efficiently with short term bans in response to fast scans lasting several seconds; while still returning a larger ban time to enable the network to be protected from slow scans lasting several days. It is assumed that the vast difference between the different types of scans means a static value will be inefficient as it will result in being one of two things: far too long for fast scans in an attempt to provide protection for slow scans, or be too short in an attempt to be efficient for fast scans. This would only result in rules being reapplied multiple times for slow scans. To verify this conclusion we included a commonly used static value as a benchmark for the dynamic calculations.

For the ban length calculation to be well suited to each source IP address it needs to be individualised to the given IP. A straightforward way to do this is to record the mean time difference between each probe from the source IP addresses. The result is that the number returned for a scan lasting several days is far larger than the resulting number from a scan which lasted under 10 seconds. This mean time interval is the foundation of the ban length calculations we tested.

Interval Squared: Squaring the interval allows for a ban length to be calculated based entirely off of the mean interval while still scaling quite high to provide protection to the network.

Interval x Interval / 2: Similar to Interval Squared, however producing a shorter ban length in an effort to possibly attaining maximum efficiency.

Interval x Threshold: While appearing to be chosen for convenience, this calculation is actually based off Figure 5 where the approximate optimum point for a interval multiplier is equal to our already existent threshold of 11.

Static 24 Hour: This value is being used a pseudo benchmark as it is sometimes used by administrators.

In addition to these calculations, a static value of 100 seconds is added to each result using the mean time interval in the calculation to allow for the cases where a source IP address probes extremely fast and the scan is completed in under a second (as in Figure 4). This results is a mean time interval of zero seconds when the ban is initially added, thus resulting in the ban being lifted as soon as it is applied.

Figure 7 illustrates the results from the different ban length calculation methods which were trialled. The y-axis details the number of iptable rules are currently in place on the network, while the x-axis details the number of iptable commands which have been sent across the network to add or remove a rule.

The twin squared ban time calculations have resulted in being the worst ban length calculations in terms of efficiency. While they are based most tightly on the time interval between attacks, resulting in the biggest difference in ban lengths, they very quickly result in bans which are in excess of what is required to stop a source IP's activities. The ban lengths which are calculated are too long and are thus efficient in terms of the number of rules being sent across the network, but the bans remain in place long after they are no longer needed. The result can be seen by the two functions lines on the graph going out of the bounds of the graph after only 5000 iptable commands have been sent.

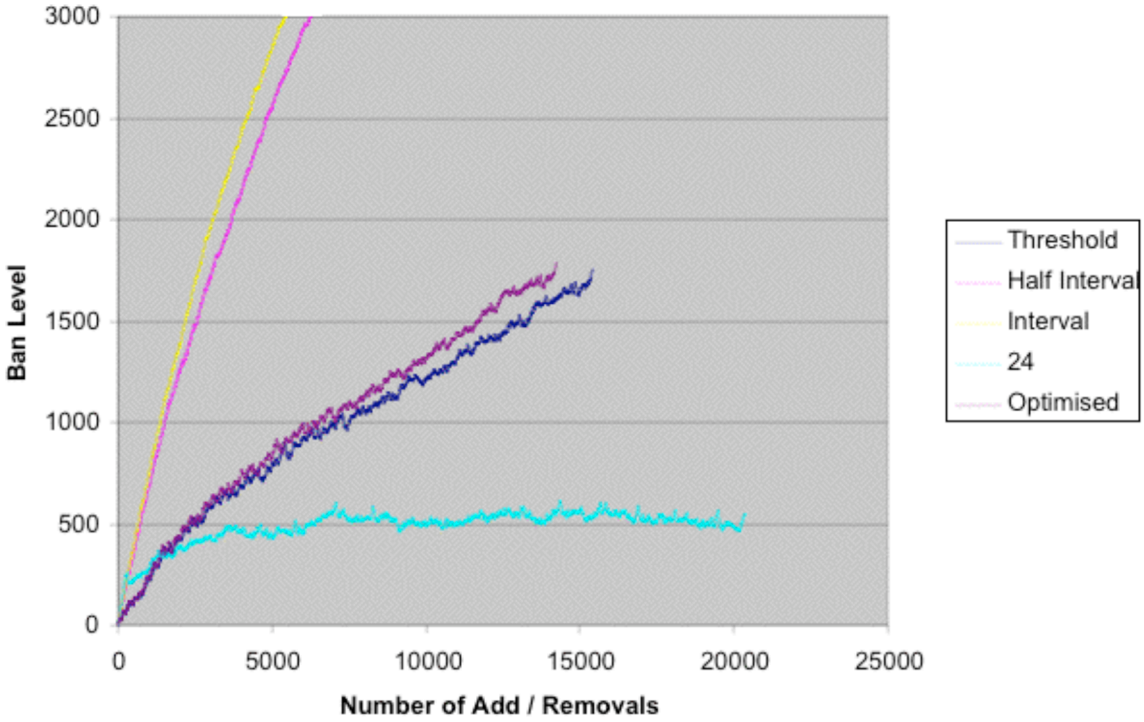

Figure 7 Comparison of Ban Length Calculation Methods

The remaining two methods, the static 24 hour ban length and the attack time interval multiplied by the threshold level each have their strengths and failings.

The static 24 hour ban produced some surprisingly good results; while initially as the worse then the optimum it quickly became the best in terms of the number of rules upon the network gateways. Now while it is obvious that the static 24 hour ban length results in having a longer ban length and longer scans require the rules to be re-added multiple times, it still performs well. This real strength of the static 24 hour ban results from it resulting in this method having something the other method do not have: a maximum ban length. Where the other methods often calculate a ban that lasts weeks or months in the worst case, it still remains at 24 hours in this method. The result is that if a ban time of a large length is seen to be needed, but the source IP actually has no interest in a long term attack and ceases their activities the rule remains on the gateways for a long time needlessly. While these ban times result from the logical scaling up of methods which work in shorter time periods; however at larger time periods the methodology which drives the dynamic ban length calculation fails. Despite returning good results in terms of the numbers of rules present on the gateways, it adds and removes rules a lot more frequently then the other methods. The 24 hour ban actually returns 50% more iptable commands then the optimised threshold method discussed shortly.

The Interval multiplied by the threshold method starts off as the best case, but results in creeping above the 24 hour static ban over the 20 day period. There are two problems one can notice when examining Figure 7: it sends more rule requests to the gateways than necessary and it keeps a lot of rules in the gateways for too long. These two problems are related in that they are the opposite of each other, and a function of the threshold method being too closely constrained by the average case.

The first problem, of sending too many rule requests to the gateways, results from ban lengths which are too short being calculated and are then lifted before the user has stopped attacking. When a ban is lifted and a source IP attacks again they get re-banned, and then eventually re-unbanned when the rule expires. The result is that 3 iptable commands are sent across the network when only one would have been required with a more accurate ban length calculation.

In an effort to counter this scenario, a version of the system was made which checked for activity by a source IP address in the previous two attack interval periods prior to the expiry time of the ban. If the source IP had been active, the ban length is increased (half threshold multiplied by interval time in this test), and no additional iptable commands are sent. The results which were produced by this were that over 1200 fewer iptable commands over the 20 day period (as seen in Figure 7 as the Optimised line); this was seen to be a significant improvement.

The larger problem with the Threshold ban length calculation is the tendency for ban times of several weeks or months to be calculated. The only method we tested where this was not an issue was the static 24 hour ban, which had the obvious maximum ban time. Therefore a maximum ban length is the obviously solution to be implemented to combat this issue; this has not been done for this paper. The maximum ban lengths which are going to be tested upon the system over the coming weeks will be 5, 7 and 10 days respectively. When examining the results from the optimised ban calculation there are 337 banned IP's out of over 1750 currently in place which have ban lengths less than 7 days. Utilizing 7 days as the

maximum ban length would cause the vast majority of currently banned source IP addresses to be examined for their validity as a current rule upon the gateways of the network. The result is expected to produce similar (slightly greater) iptable rule count then the 24 hour method with still having lower numbers of rules being sent across the network then the 24 hour method.

In addition to the system storing a state of the network in terms of current banned source IP's, the Action module also keeps a record for a short time of past banned users. This enables the system to have knowledge of a malicious IP without needing to have the IP blocked at the gateway. This longer term retention of contextual information allows for the analysis module to recognise an IP who has been banned previously to be re-banned by the action module without needing to again be analysed as being a threat. This 'safety net' allows for additional protection against attackers who do not fit the model of behaviour which suits most source IP addresses.

The Ban History, Banned, and Audit tables in the database are regularly cleaned of expired and old entries to keep the database size as small as practicable. This prevents any performance gain at the firewall due to the efficiency of the rules from being lost by a overly costly centralised analysis process.

5 Related Work

There have been continuous developments and advancements in the examination of electronic audit logs since the 1980's. However this has only rarely branched into the realm of amalgamating logs across gateways to examine threats at a network wide level. Recent research in this area has occurred however with the MINDS project.

5.1 MINDS

The MINDS (2004) (Minnesota Intrusion Detection System) project has the objective of producing a system which will allow large scale analysis using data mining algorithms to detect attacks (MINDS 2004). The MINDS system uses a combination of signature detection and anomaly detection to provide protection to the University of Minnesota network.

The MINDS system uses network traffic flow data collected from CISCO routers. This audit data is then filtered to remove extraneous entries before feature extraction collates the required information for analysis (source and destination IP's and ports, protocols, timestamp, flags). Also catalogued is derived contextual information such as the amount of traffic to a destination from a specific source. The extracted, reduced log is then run through the Attack Detection Module of MINDS using signature detection to discover any known attacks. The remaining log is then fed through the Anomaly Detection Modules that allocates a score to each connection in relation to normal traffic patterns. Connections that score highly are then further analysed by the network administrators to moderate whether or not the connection was an intrusion or a false positive. Connections that scored highly by the Anomaly Module, and are not found to be false positives by the administrators, are then further analysed to produce new signatures for emerging attacks. It is in this way the MINDS system is able to not only protect against the more common and well known attacks, but is also very strong on the detection of novel attacks, or attacks which are not yet supported by many other IDS (Ertoz et al. 2003; Ertoz et al. 2004).

The MINDS project has been developed during the same period as our own system, and as such, there are some notable differences between the two implementations. The MINDS project does not employ threshold level heuristics in their detection mechanisms, and the system is not fully intended to be automated process. As it is does not deal with malicious source IP addresses as an automated process, it also does not dynamically calculate how long to ban an IP based on their activity; but requires an administrator to take the action needed.

6 Further Work

The implementation of our system has progressed well from being able to initially detect source IP addresses probing multiple gateways to being able to ban them efficiently from the network. As a result of being able to carry out the action needed on the main attack of interest the continuing work on the system will focus on other modes and types of attack which can occur across multiple gateways.

The system currently cleans out the Ban History and Audit tables as the IP addresses of attackers frequently change. Further work for the system is to find a way to use or enlarge the source IP profile information to be able to link different source IP activity to being a single user, thus allowing for action to be taken against them faster.

The system currently records whether or not an attack has been across multiple ports, and then can track the ports using the Tracking module. Often attacks which occur across multiple gateways also occur across multiple ports on those gateways, examining the data relating to this could increase the rate of detection of threats against the network.

7 Conclusion

Phase 1 of our system showed that it was possible to detect malicious source IP addresses which were probing multiple gateways connected upon the same network. Previously such trivial attacks have been overlooked by network security infrastructure, always examining incoming packets in the context of a single gateway, exposing networks to a broader more methodical attack.

Phase 2 allows for such attacks to not only be detected but to also be dealt with through the creation of an Action module which sends out iptable rules to the relevant gateways upon the network, with the aim of providing the needed protection prior to the attacks 'arrival' at the vulnerable gateways.

The Action module not only creates the necessary rules, but also removes the bans once they have expired according to the ban time calculation made by the action

module. The result is an efficient attack detection system, aiming to provide protection to ever growing and expanding private networks.

8 References

Brox, A 2002, 'Signature Based or Anomaly Based Intrusion Detection – The Practice and Pitfalls', *Schmagazine*.

Cheswick, WR, Bellovin, SM & Rubin, AD 2003, *Firewalls and Internet Security Second Edition*, Second edn, Professional Computing Series, Addison-Wesley, Boston.

Ertoz, L, Eilertson, E, Lazarevic, A, Tan, P, Dokas, P, Srivastava, J & Kumar, V 2003, *Detection and Summarization of Novel Network Attacks Using Data Mining*.

Ertoz, L, Eilertson, E, Lazarevic, A, Tan, P, Srivastava, J, Kumar, V & Dokas, P 2004, 'The MINDS - Minnesota Intrusion Detection System', in *Next Generation Data Mining*.

Heberlein, L, Dias, G, Levitt, K, Mukherjee, B, Wood, J & Wolber, D 1990, 'A Network Security Monitor', *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp. 296-304.

Hoffman, D, Prabhakar, D & Strooper, P 2003, 'Testing iptables', paper presented to IBM Centre for Advanced Studies Conference, Toronto, Ontario, Canada.

Hofmeyr, S, Forrest, S & Somayaji, A 1998, 'Intrusion Detection using Sequences of System Calls', *Journal of Computer Security*, vol. 6, p. 151/80.

Holden, G 2003, *Guide to Network Defence and Countermeasures*, Course Technology, Thomson.

Kumar, S 1995, 'Classification and detection of computer intrusions.' PhD thesis, Purdue University.

MINDS, RT 2004, *MINDS - Minnesota Intrusion Detection System*, <http://www.cs.umn.edu/research/minds/MINDS.htm>.

Porras, P & Neumann, P 1997, 'EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances', paper presented to National Information Systems Security Conference, Baltimore, MD.

Scanlan, J, Lorimer, S, Hartnett, J & Manderson, K 2004, *Intrusion Detection by Intelligent analysis of data across multiple gateways in real-time*, <http://eprints.comp.utas.edu.au:81/archive/00000049/>.

Sommer, R & Paxson, V 2003, 'Enhancing byte-level network intrusion detection signatures with context', paper presented to Conference on Computer and Communications Security, Washington D.C.

Vigna, G, Eckmann, ST & Kemmerer, RA 2000, 'The STAT Tool Suite', paper presented to DISCEX 2000, Hilton Head, South Carolina.